IN THE CLAIMS

1. (Currently Amended) A method in a data processing system having a program for a server to handle one or more client requests, the method comprising the steps of:

obtaining one or more of said client requests for hierarchically organized data at a server;

dividing said client requests into one or more smaller units, each of said smaller units being a transaction request serviceable by one of a plurality of worker threads, each of said smaller units including an envelope having a beginning tag and an ending tag that are recognizable by the respective worker threads;

placing said smaller units in a queue; and

servicing said units in order, the worker threads stopping service of a respective smaller unit upon encountering an ending tag.

2. (Original) The method of claim 1 wherein said client requests are in XML format.

3. (Original) The method of claim 1 wherein said hierarchically organized data is stored using a Document Object Model.

4. (Canceled)

5. (Original) The method of claim 1 wherein said server is a registry server.

6. (Previously Presented) The method of claim 1 wherein said queue is handled using a FIFO scheduling algorithm.

7. (Currently Amended) The method of claim 1 wherein said smaller units are defined by an XML <envelope> and an XML </envelope>tag.

8. (Currently Amended) A computer program product comprising:

a computer usable medium having computer readable program code embodied therein configured to cause a server to handle one or more client requests comprising:

computer readable code configured to cause a computer to obtain one or more of said client requests for hierarchically organized data at a server;

computer readable code configured to cause a computer to divide said client requests into one or more smaller units, each of said smaller units being a transaction request serviceable by one of a plurality of worker threads, each of said smaller units including an envelope having a beginning tag and an ending tag that are recognizable by the respective worker threads;

computer readable code configured to cause a computer to place said smaller units in a queue; and

computer readable code configured to cause a computer to service said units in order, the worker threads stopping service of a respective smaller unit upon encountering an ending tag.

9. (Original) The computer program product of claim 8 wherein said client requests are in XML format.

10. (Original) The computer program product of claim 8 wherein said hierarchically organized data is stored using a Document Object Model.

11. (Canceled)

12. (Original) The computer program product of claim 8 wherein said server is a registry server.

13. (Previously Presented) The computer program product of claim 8 wherein said queue is handled using a FIFO scheduling algorithm.

14. (Currently Amended) The computer program product of claim 8 wherein said smaller units are defined by an XML <envelope> and an XML </envelope>tag.

15. (Currently Amended) A server framework in a computer system comprising:

a memory for storing one or more client requests for hierarchically organized data from a server; a thread pool object configured to divide said requests into one or more smaller units,

each of said smaller units being a transaction request serviceable by one of a plurality of worker threads, each of said smaller units including an envelope having a beginning tag and an ending tag that are recognizable by the respective worker threads; placing said smaller units in a queue; and one or more worker objects configured to service said units in order, the worker threads stopping service of a respective smaller unit upon encountering an ending tag.

16. (Original) The server framework of claim 15 wherein said client requests are in XML format.

17. (Original) The server framework of claim 15 wherein said hierarchically organized data is stored using a Document Object Model.

18. (Canceled)

19. (Original) The server framework of claim 15 wherein said server is a registry server.

20. (Previously Presented) The server framework of claim 15 wherein said queue is handled using a FIFO scheduling algorithm.

21. (Currently Amended) The server framework of claim 15 wherein said smaller units are defined by an XML <envelope> and an XML </envelope>tag.

22. (Currently Amended) A system for implementing a server framework comprising:
a processor; and
 a memory including:
        one or more requests for hierarchically organized data transmitted from a client to a server;
        a thread pool object configured to divide said requests into one or more smaller units, each of said smaller units being a transaction request serviceable by one of a plurality of worker objects, each of said smaller units including an envelope having a beginning tag and an ending tag that are recognizable by the respective worker objects;

a queue in which said smaller units are placed; and

one or more worker objects configured to service said units in order.

placing said smaller units in a queue; and

servicing said units in order, the worker objects stopping service of a respective smaller unit upon encountering an ending tag.

23. (Original) The system of claim 22 wherein said requests are in XML format.

24. (Original) The system of claim 22 wherein said hierarchically organized data is stored using a Document Object Model.

25. (Canceled)

26. (Original) The system of claim 22 wherein said server is a registry server.

27. (Previously Presented) The system of claim 22 wherein said queue is handled using a FIFO scheduling algorithm.

28. (Currently Amended) The system of claim 22 wherein said smaller units are defined by an XML <envelope> and an XML </envelope>tag.

29-35. (Canceled)

36. (Previously Presented) The method of claim 1 wherein a plurality of client requests are received and units from the plurality of client requests are placed in the same queue.

37. (Previously Presented) The computer program product of claim 8 wherein a plurality of client requests are received and units from the plurality of client requests are placed in the same queue.

38. (Previously Presented) The server framework of claim 15 wherein a plurality of client requests are received and units from the plurality of client requests are placed in the same queue.

39. (Previously Presented) The system of claim 22 wherein a plurality of client requests are received and units from the plurality of client requests are placed in the same queue.

40. (Previously Presented) The method of claim 36 wherein the plurality of client requests are received through a plurality of sockets, and wherein the worker thread services units received through at least two of the plurality of sockets.

41. (Previously Presented) The computer program product of claim 37 wherein the plurality of client requests are received through a plurality of sockets, and wherein the worker thread services units received through at least two of the plurality of sockets.

42. (Previously Presented) The server framework of claim 38 wherein the plurality of client requests are received through a plurality of sockets, and wherein the worker thread services units received through at least two of the plurality of sockets.

43. (Previously Presented) The system of claim 39 wherein the plurality of client requests are received through a plurality of sockets, and wherein the worker object services units received through at least two of the plurality of sockets.